

WHAT IS CLAIMED IS:

1 1. A partial merging method for sub-register data operations in a processor, the method
2 comprising:

3 examining an instruction before execution to identify a portion of a source register
4 identified in the instruction that should remain unchanged into a destination register; and
5 moving the portion of the source register determined to remain unchanged into the
6 destination register before instruction execution is complete.

1 2. The method of claim 1, wherein moving the unchanged portion of the source register into
2 the destination register includes setting corresponding source register values to zero.

1 3. The method of claim 1, wherein the source and destination registers have a greater bit-
2 length than a result bit-length of the instruction.

1 4. The method of claim 3, wherein the bit-length of the source and destination registers is 32
2 bits.

1 5. The method of claim 3, wherein the result bit-length is greater than 1 bit and less than 32
2 bits.

1 6. The method of claim 3, wherein the result bit-length is greater than 1 bit and less than 16
2 bits.

1 7. A method for sub-register data operations for executing an instruction, the method
2 comprising:

3 executing the instruction on a first register and a second register; and
4 merging a result of the executed instruction with a plurality of high-order bits from the first
5 register, the plurality of high-order bits being copied into high-order bit positions of a result register,
6 and the result being placed into low-order bit positions of the result register.

1 8. The method as recited in claim 7, wherein the merging a result comprises:
2 modifying contents of the second register by placing data values of zero in the high-order bit
3 positions of the second register;
4 adding contents of the first register with the modified second register; and
5 placing the result in the result register.

1 9. The method as recited in claim 8, the method further comprising:
2 ignoring a carryover of the result from the low-order bit positions of the result register to the
3 high-order bit positions of the result register.

1 10. The method as recited in claim 7, wherein the merging a result comprising:
2 modifying contents of the first register by placing data values of zero in the low-order bit
3 positions of the first register;
4 modifying contents of the second register by placing data values of zero in the high-order bit
5 positions of the second register;
6 adding the modified first register with the modified second register; and
7 placing the result in the result register.

1 11. The method as recited in claim 10, the method further comprising:
2 ignoring a carryover of the result from the low-order bit positions of the result register to the
3 high-order bit positions of the result register.

1 12. The method of claim 7, wherein the first register and the second register have 32 bits.

1 13. The method of claim 7, wherein the result register has 32 bits.

1 14. The method of claim 7, the method further comprising:
2 using a renamer to assign the first register, the second register, and the result register.

1 15. The method of claim 7, wherein the result of the executed instruction is less than 32 bits.

1 16. The method of claim 15, wherein the result of the executed instruction is less than or equal
2 to 16 bits.

1 17. The method of claim 16, wherein the result of the executed instruction is less than or equal
2 to 8 bits.

1 18. The method of claim 7, wherein the merging a result is performed before instruction
2 execution is complete.

1 19. A processor comprising:
2 an instruction set having an instruction;
3 a source register and a destination register referenced by the instruction from the instruction
4 set; and
5 a logic circuit adapted to examine the instruction before execution to identify a portion of the
6 source register that should remain unchanged into the destination register, and the logic circuit
7 further adapted to move the unchanged portion into the destination register before instruction
8 execution is complete.

1 20. The processor of claim 19, wherein the logic circuit is adapted to move the unchanged
2 portion into the destination register by setting corresponding values of the source register to zero.

1 21. The processor of claim 19, wherein the source register and the destination register have a
2 greater bit-length than a result of the instruction.

1 22. The processor of claim 21, wherein the source register and the destination register have 32
2 bits.

1 23. The processor of claim 21, wherein the result of the instruction has less than 32 bits.

1 24. The processor of claim 23, wherein the result of the instruction is less than or equal to 16 bits.

1 25. A method for executing instructions in a processor using data registers of different bit lengths
2 while maintaining architecture compatibility, the method comprising:
3 receiving an instruction to perform an operation on contents of first and second source
4 registers, the contents including a plurality of bits and the operation results being a different bit
5 length then bit lengths of the first and second source registers;
6 screening the first and second source registers; and
7 merging the operation results into a destination register when the operation is performed.

1 26. The method of claim 25, wherein screening the first and second source registers
2 comprising:
3 identifying high order bits of one of the source registers that should remain unchanged
4 when merged into the destination register; and
5 modifying the contents of the other source register by setting corresponding high order
6 bits of the other source register to zero.

1 27. The method of claim 26, wherein merging the operation results comprising:
2 adding the contents of the one of the source registers with the modified contents of the
3 other source register; and
4 placing results of the addition in the destination register.

1 28. The method of claim 27, further comprising:
2 ignoring a carryover of the addition results from low-order bit positions of the destination
3 register to high-order bit positions of the result register.

1 29. The method of claim 25, wherein screening the first and second source registers
2 comprising:
3 modifying contents of one of the source register by setting low order bits of the one of the
4 source registers to zero; and
5 modifying the contents of the other source register by setting high order bits of the other
6 source register to zero.

1 30. The method of claim 29, wherein merging the operation results comprising:
2 adding the modified contents of the one of the source registers with the modified contents
3 of the other source; and
4 placing results of the addition into the destination register.